

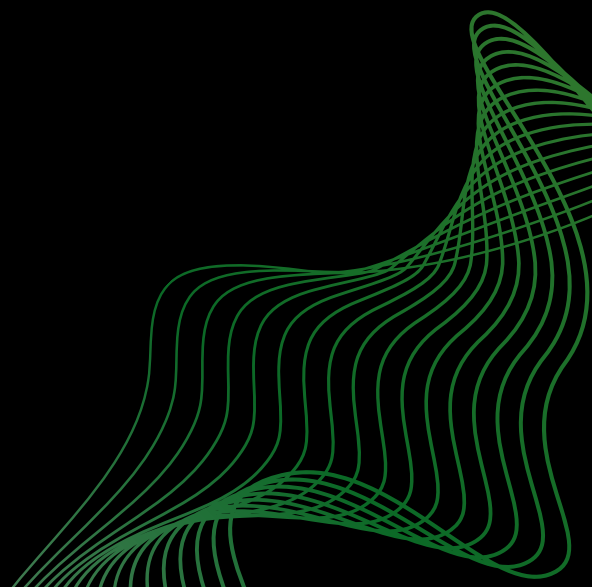


MIDGARD

Cardano Optimistic Rollup L2

built by Anastasia Labs

Project Scope



Contents

1. Mission and Vision
2. Executive summary
 - 2.1. Disclaimer: proposal assumptions and bias
3. Why Rollups
4. Comparison to similar projects
5. Estimates and contributions
 - 5.1. Overall sizing
 - 5.2 Already secured resources
 - 5.3 Final ask
6. Sizing and managing the project
 - 6.1. Midgard framework: turning vision into achievable goals
 - 6.2. Parallel Development
 - 6.3 Architecture
7. Delivering Midgard
 - 7.1. Onchain Development
 - 7.2. Offchain SDK Development
 - 7.3 Midgard Layer 2 Node Development
 - 7.4. Tx Builder Modification for Midgard Layer 2
 - 7.5. Testnet Deployment - DevOps/Distributed System Specialist
 - 7.6. Data Availability (DA) Layer Implementation
8. Conclusion



MIDGARD



1. Mission and Vision

Midgard aims to revolutionize Cardano's scalability and economic sustainability by introducing the first optimistic rollup framework. Our mission is to enhance transaction throughput, reduce costs, and enable complex decentralized applications, all while maintaining Cardano's core principles of security and decentralization.

As a Layer 2 (L2) framework, Midgard extends the capabilities of Cardano, facilitating high-throughput low fee transactions. Its innovative design aims to provide a clear avenue towards economic sustainability for Cardano by driving significant fee revenue without increasing transaction fees for regular users of the L1. Midgard rollups can afford to pay higher transaction fees for block and blob space because these costs are distributed across the many transactions within a rollup block. Through its modularity Midgard aims to empower developers with the ability to design and deploy rollups with their own tradeoff profiles tailored specifically for their domain without requiring changes to the Cardano L1. For instance, if your application would benefit from fee markets, programmable native assets, or protocol level support for real world assets, DIDs, or soulbound tokens, you can deploy your application on a Midgard rollup with those features and benefit from the user activity, functionality, and security of Cardano. No single network can perfectly suit the needs of every possible application, with Midgard, applications can deploy to the sub-network best suited to their domain requirements without losing the functionality and security of Cardano. With community collaboration and a focus on interoperability, Midgard positions Cardano as a leader in blockchain innovation, delivering a scalable, economically sustainable, and resilient ecosystem for the future.

The primary drivers behind the design of Midgard are the goals of enhancing scalability, advancing broad adoption, and ensuring the economic sustainability of Cardano, without compromising on decentralization or security.



MIDGARD

2. Executive summary

Midgard is a pioneering Layer 2 (L2) scaling framework for the Cardano blockchain, designed to enhance transaction throughput, reduce costs, and enable advanced decentralized applications. Utilizing optimistic rollup technology, Midgard achieves scalability without compromising the security and decentralization principles that define Cardano.

By offloading transaction processing from Layer 1 (L1) to L2, Midgard significantly boosts Cardano's capacity to handle a higher volume of transactions, supporting the growing demand for more complex applications. Key features include efficient fraud-proof mechanisms, a robust operator management system, and a modular design that facilitates easy integration and innovation within the Cardano ecosystem.

Midgard, by default, has a tokenless design. All transaction fees on Midgard are paid in ADA, and every Midgard block is published to the Cardano L1. Where Hydra moves transactions offchain, and thus does not contribute consistently in fee revenue to the Cardano L1, Midgard keeps transactions onchain but in a more compact form (i.e. rolls up transactions offchain into a compact representation that is published onchain). This means that increased economic activity on Midgard directly translated to increased economic activity on Cardano, this represents a significant potential source of fee revenue for the Cardano L1. Midgard does not have independent consensus or economic security, instead it inherits economic security and consensus from Cardano. The project is built on a strong foundation of collaboration with contributors from leading organizations in the Cardano ecosystem, including Anastasia Labs, Fluid Tokens, and IOG. With a phased development plan, Midgard aims to deliver an L2 solution that enhances user experience and developer capabilities while maintaining Cardano's commitment to security and decentralization.

To achieve its ambitious goals, Midgard is seeking \$1,505,000 in funding. This funding will enable the team to advance development, deploy a series of milestone-based demos, and deliver a mainnet-ready solution by the end of 2025.



MIDGARD



2.1. Disclaimer: proposal assumptions and bias

We acknowledge the challenges inherent in developing the first framework for building modular L2s on Cardano. To provide transparency, we highlight the following assumptions and considerations:

- **Ground-Up Development with Strong Foundations:** Midgard is being built from scratch, as no existing rollup frameworks currently exist on Cardano. Despite this, we have already completed a comprehensive technical specification and made significant progress on implementing key components, including the state queue, operator directory, and fraud proof mechanisms. This head start positions us to achieve our ambitious development goals within the proposed timeline.
- **Economic Incentives as a Driver:** Midgard assumes that L2 protocols like itself will significantly contribute to Cardano's economic sustainability by paying competitive transaction fees for block and blob space. These fees are distributed across rollup transactions, making them cost-effective while driving fee revenue to the network.
- **Phased Rollout for Clear Progress:** Midgard will be delivered in stages, with each component designed, implemented, and demonstrated to showcase incremental progress. Key milestones, such as the development of the state queue, and integration of fraud proof mechanisms, will provide clear evidence of progress toward our ultimate goal of mainnet readiness.
- **Commitment to Open Source Excellence:** Midgard is built with an open-source ethos, leveraging our proven track record of delivering high-quality, widely adopted tools such as Lucid Evolution and Plug-and-play Smart Contracts. These contributions have become cornerstones of the Cardano ecosystem, showcasing our ability to create robust, user-friendly solutions that accelerate adoption and innovation.



MIDGARD

Our goal is to achieve production readiness for the Midgard framework by the third quarter of 2025, enabling seamless deployment of fully functional optimistic rollups on the Cardano Mainnet.

This milestone will mark the framework's maturity, providing developers with the tools to efficiently spin up scalable Layer 2 solutions and unlock new possibilities for decentralized applications on Cardano. We attempted to be very conservative with this timeline, however, we acknowledge that given we are working in uncharted territory this could be overly optimistic and we may encounter unforeseen challenges. As an extremely conservative estimate, we have extreme confidence that at the very least the framework will be testnet ready by the end of the aforementioned timeline.



MIDGARD



3. Why Rollups

We have Hydra as our native state of the art state channel solution, we have partnerchains as our native evolution of sidechains, Midgard aims to be our native evolution of rollups.

The UTxO system is a match made in heaven for rollups. It allows us to build true L2 rollups that inherit maximal security from Cardano. This cannot be replicated in account based systems. It's not a coincidence that Fuel, the first general purpose L2 to receive a decentralization rating of stage 2 (highest possible) is UTxO based.

Fraud proofs for global state systems like Arbitrum and Optimism are extremely difficult to implement and very expensive and complicated to conduct onchain (and require multiple parties); this is why to this date despite spending millions in R&D neither of those protocols has working fraud proofs, and they all rely on centralized permissioned sequencers and operators.

On the other-hand, fraud proofs on Cardano are extremely straightforward and require only a single party (no challenge-response proofs) due to the local state properties of the ledger.



MIDGARD

Not a single blockchain has managed to achieve true permissionless general purpose rollups. At the end of the day, the top “L2s” in the blockchain space right now are all custodial multisigs. The Midgard framework is a first of its kind, in its capabilities to deploy completely permissionless rollups that inherit the full security of Cardano.

- No centralized sequencer
- No challenge response “proofs”
- No custodial multisig
- Permissionless fraud proofs (open to anyone)
- Permissionless operator set
- Deposits and withdrawals cannot be censored (inherits the full censorship resistance of Cardano)

This is all made possible by Cardano’s unique local state EUTxO architecture.

UTxO contention, small block size, local state, transaction determinism, all of these “problems” may have led you to question why these design choices were made.

Midgard aims to show you that these are actually not problems at all. In-fact quite the opposite, they are extremely powerful properties that, together, provide unique value that simply does not exist in any other ecosystem. These are, in actuality, the core pillars that make Midgard even possible in the first place.

You cannot build Midgard on Ethereum, Solana or Sui. It is a protocol that is only possible on Cardano.



MIDGARD



4. Comparison to similar projects

Optimism

Project	Optimism
Description	An Ethereum Layer 2 scaling solution utilizing optimistic rollups to improve throughput and lower transaction fees while preserving the decentralized security of the Ethereum network.
Timeline	3 years of development before mainnet
Dedicated team	~8 core developers; dozens of external contributors
Estimated initial cost	>3.5M\$

Arbitrum

Project	Arbitrum
Description	A Layer 2 solution for Ethereum using optimistic rollups to scale transactions off-chain, reducing costs and congestion while maintaining security through Ethereum's base layer.
Timeline	2 years of development before mainnet
Dedicated team	~7 core developers; ~150+ external contributors
Estimated initial cost	3.7M\$

Fuel

Project	Fuel
Description	Fuel is a modular Layer 2 scaling solution for Ethereum using a UTXO model. Powered by FuelVM, it offers high throughput, low costs, and parallel transaction execution for efficient scaling.
Timeline	2 years of development before mainnet
Dedicated team	~5 core developers; ~50 external contributors
Estimated initial cost	1.5M\$

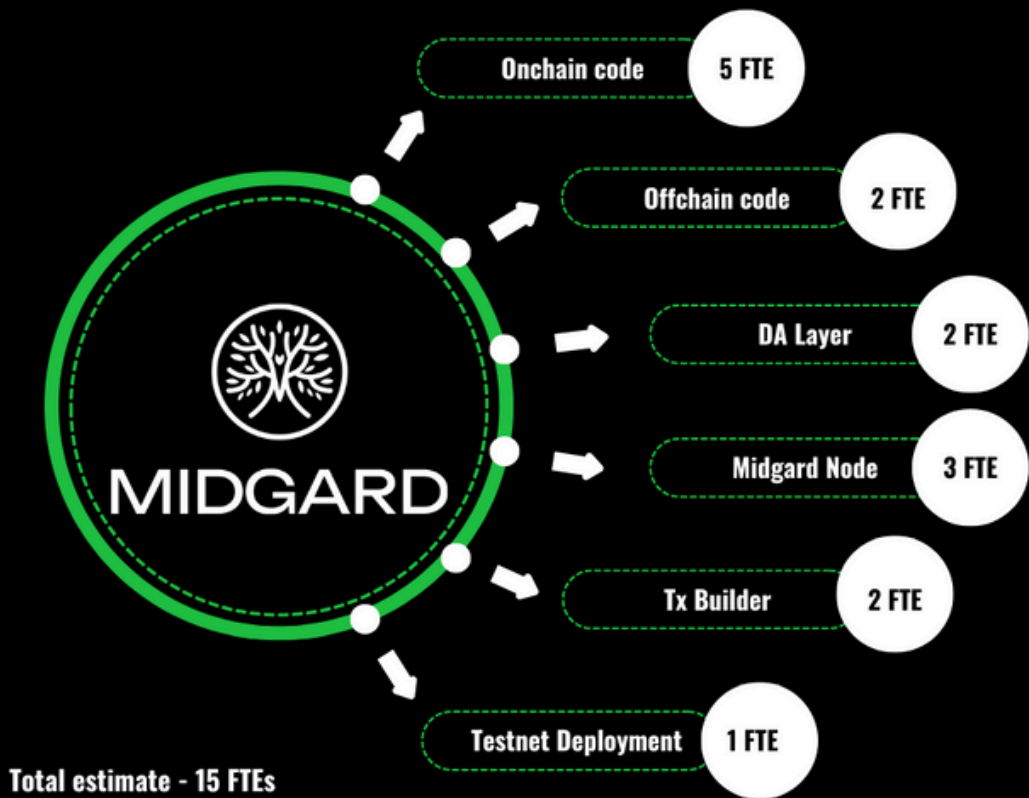


MIDGARD

5. Estimates and contributions

5.1. Overall sizing

Midgard - Scopes overview: sizing the work (2025 yearly basis)

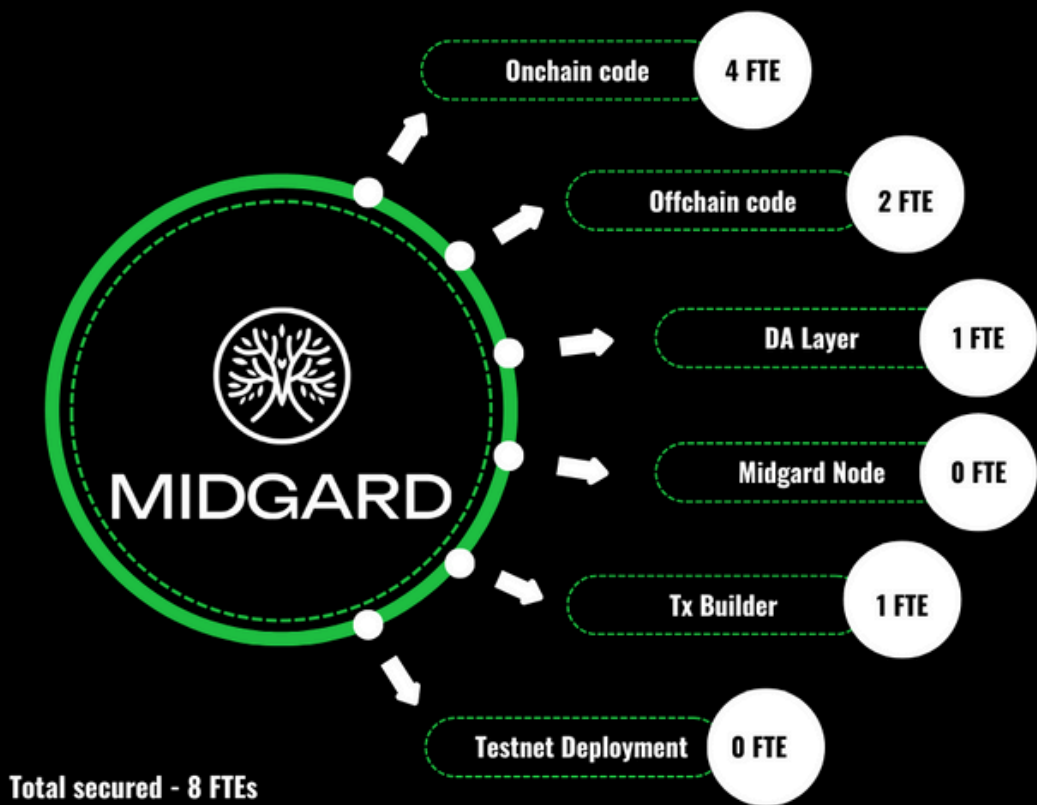


Scope	Resources estimated
Onchain	5 FTEs
Offchain	2 FTEs
Data Availability Layer	2 FTEs
Midgard Node	3 FTEs
Tx Builder	2 FTEs
Testnet Deployment	1 FTEs

Total estimated for 2025 delivery: 15 FTEs .

5.2. Already secured resources

Midgard - Scopes overview: resources secured (2025 yearly basis)



We have invested ~\$1M of our own funds, and with that we have secured the following resources:

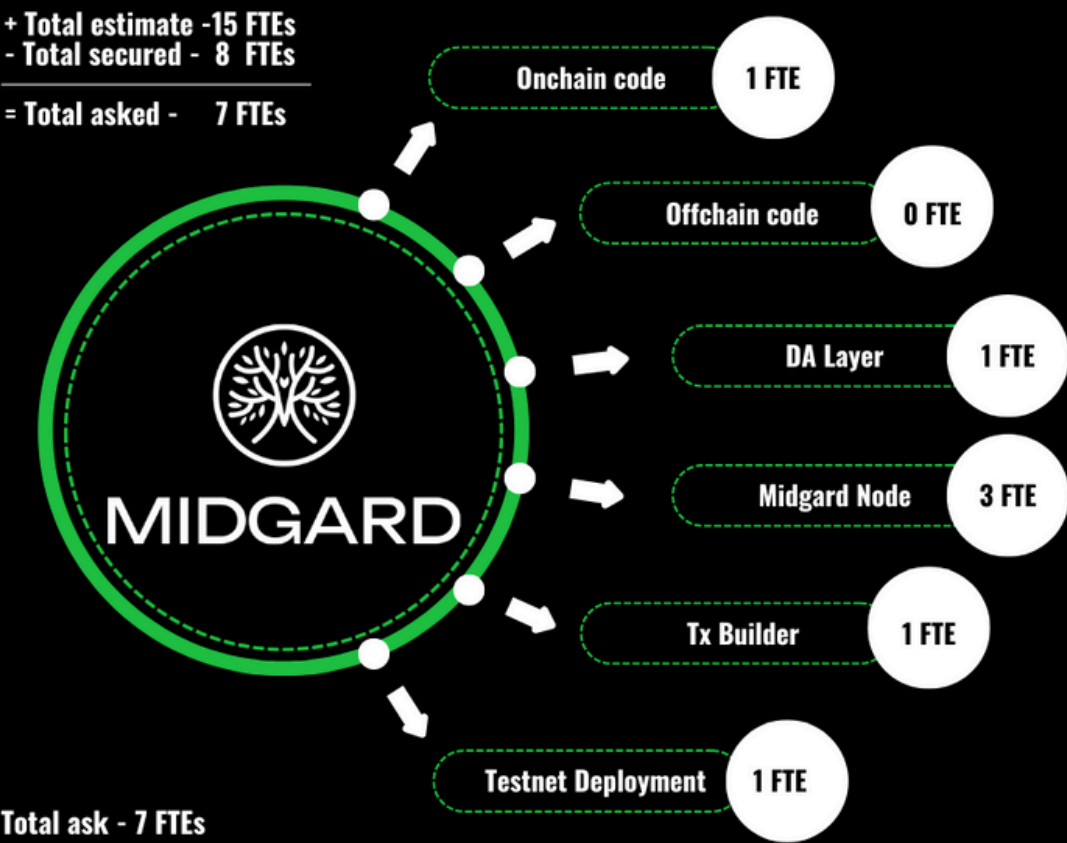
Scope	Resources already secured
Onchain	4 FTEs
Offchain	2 FTEs
Data Availability Layer	1 FTEs
Midgard Node	0 FTEs
Tx Builder	1 FTEs
Testnet Deployment	0 FTEs

Total secured resources for 2025 delivery: 8 FTEs.

5.3. Final ask

Midgard - Scopes overview: resources asked (2025 yearly basis)

+ Total estimate -15 FTEs
- Total secured - 8 FTEs
= Total asked - 7 FTEs



Here are the funds we ask for each scope:

Scope	Resources estimated	Resources secured	Resources asked
Onchain	5 FTEs	4 FTEs	1 FTEs
Offchain	2 FTEs	2 FTEs	0 FTEs
Data Availability Layer	2 FTEs	1 FTEs	1 FTEs
Midgard Node	3 FTEs	0 FTEs	3 FTEs
Tx Builder	2 FTEs	1 FTEs	1 FTEs
Testnet Deployment	1 FTEs	0 FTEs	1 FTEs

Total of the asked for our proposal: 7 FTEs

Given our assumptions the valuation of this proposal based on a 215.000\$ comes down to: $7 * 215k\$ = \$1,505.000$

Scope	Resources estimated
Onchain	\$215.000
Offchain	\$0
Data Availability Layer	\$215.000
Midgard Node	\$645.000
Tx Builder	\$215.000
Testnet Deployment	\$215.000

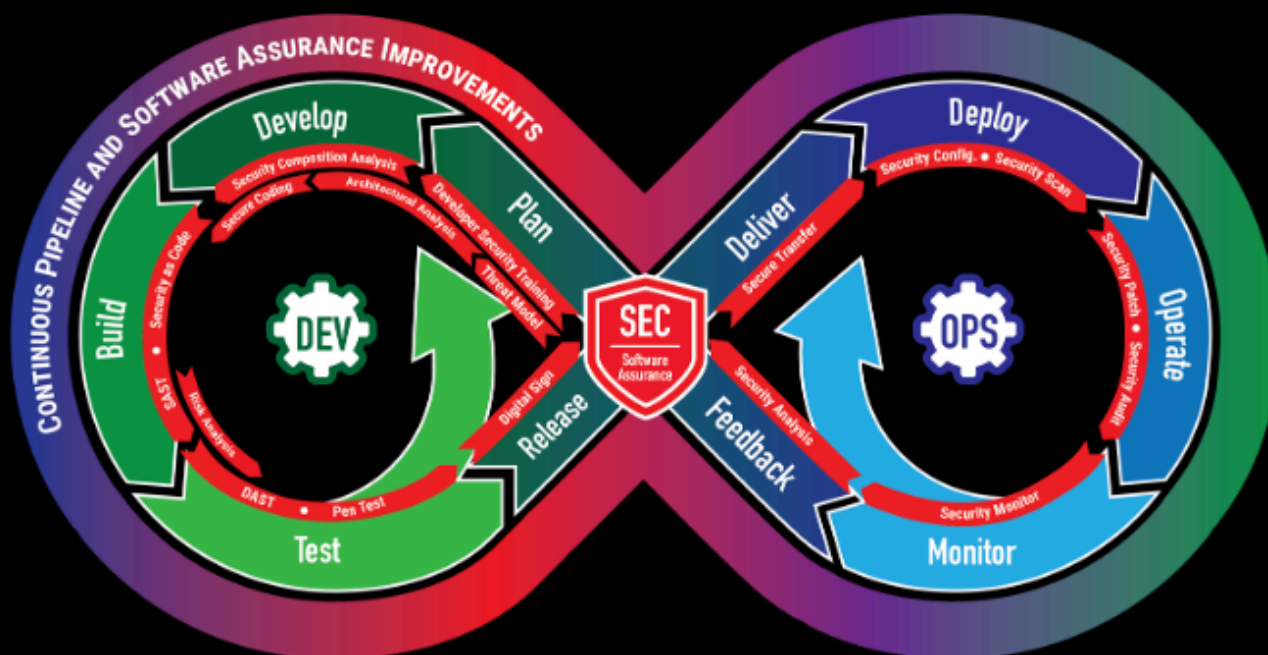
6. Sizing and managing the project

6.1. Midgard framework: turning vision into achievable goals

The key challenge in building the Midgard Framework is ensuring that Midgard Rollups inherit Cardano's full security. This requires the fraud-proof system to be comprehensive, robust and extremely efficient, and requires that all L2 block data that is needed to create fraud proofs is available in the Data Availability (DA) Layer.

Additionally, ensuring native interoperability between all Midgard Rollups and ensuring the liveliness of the Midgard Node are crucial.

The framework consists of many complex components. However, each component can be developed in parallel, allowing different teams to focus on specific tasks in their own sprints with clear objectives. To ensure timely delivery, we are adopting a DevOps lifecycle, similar to the approach used by leading tech companies..



Continuous Integration and Continuous Delivery (CI/CD) Development cycle

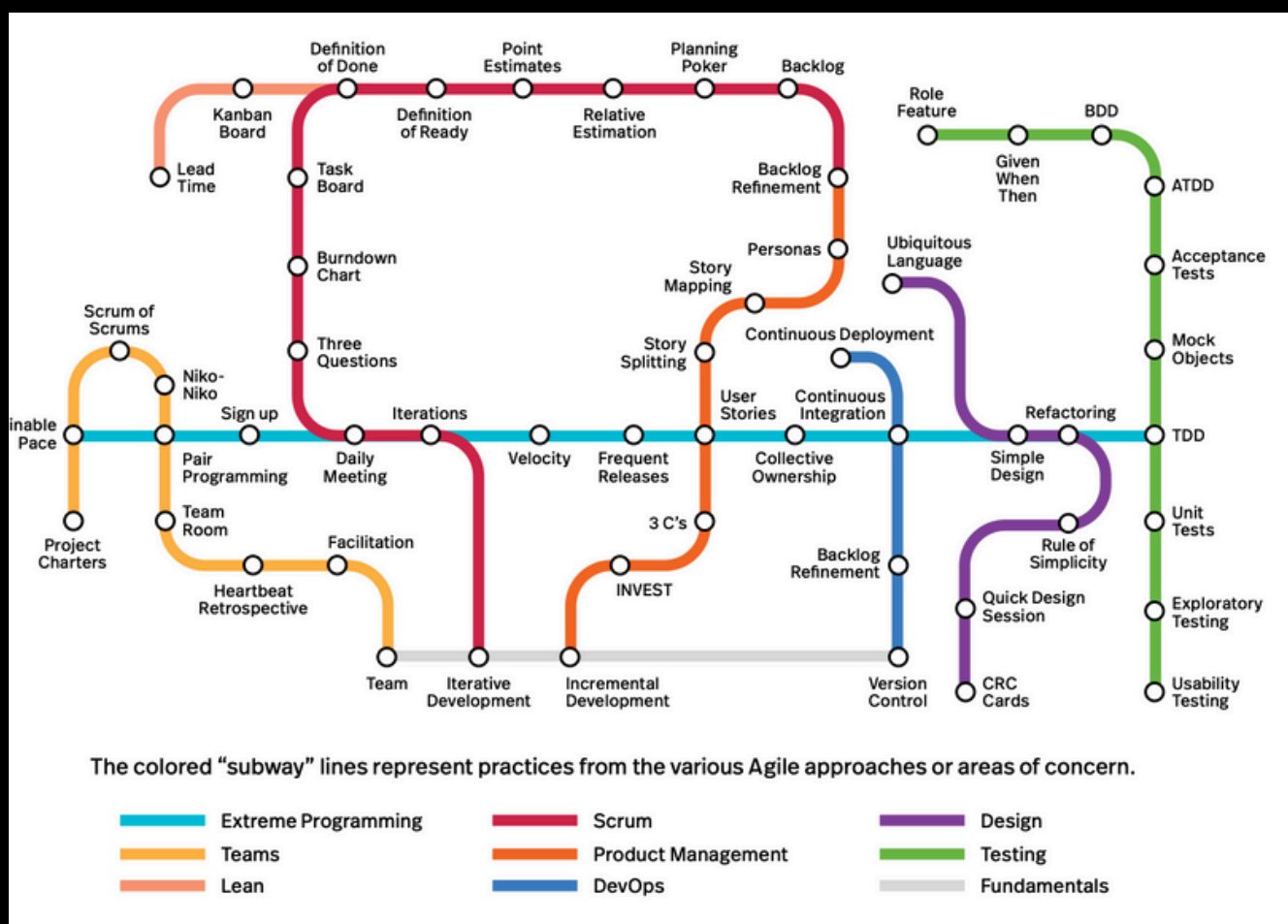


MIDGARD

6.2. Parallel Development

Most components can be developed independently in parallel up until the integration phase. Once the core components are ready, the focus will shift to integrating them into a unified Layer 2 framework.

We are following an Agile methodology, recognizing that this project has multiple phases—research, development, learning, and ultimately, delivering a functional product



Agile practices.



6.3. Architecture

The architecture and design specification for the Midgard Layer 2 framework has been meticulously crafted to take advantage of the full potential of Cardano. We have dedicated a significant amount of time to research and design, ensuring that every component of the Layer 2 network is well thought out and hyper-optimized.

As part of this process, we have compiled an extensive technical specification detailing every aspect of the architecture, including the interactions between the on-chain contracts, off-chain components, and the Data Availability (DA) layer. The specification also covers the transaction structure, security mechanisms, deposit/withdrawal protocols, and the fraud-proof system that will govern the network.

In addition to extensive internal research, we have collaborated with key players in the Cardano ecosystem, such as IOG and core Cardano developers, to validate the implementation and design specifications. These validations ensure that our approach aligns with industry standards guaranteeing the robustness and reliability of the Midgard Layer 2 framework.

You can find the technical specification of the framework on our GitHub repository, available at:

<https://github.com/Anastasia-Labs/midgard/tree/main/technical-spec>



MIDGARD



The Midgard Layer 2 Architecture consists of several interconnected components, each with specific roles that interact with both Layer 1 and Layer 2.

- Onchain development
- Offchain development
- Midgar Node
- Data Availability Layer
- Tx Builder
- Wallet Integration

MIDGARD MVP

Onchain code

- Directory linked list
- Midgard State queue
- Deposit/Withdrawal
- Fraud proof

Offchain code

- Directory linked list
- Midgard State queue
- Deposit/Withdrawal
- Fraud proof

DA Layer

- Database -> postg
- Tables
- Based on DBSync tables

Midgard Node

- NodeJS
- Config Files
- RPC
- API server
- API endpoint -> submitTx

Tx Builder

- Lucid Evolution
- Constraint tx builder
- newUTxO
- newMidgardTx().collect

Wallet Integration

- Lace wallet integration



MIDGARD

7. Delivering Midgard

7.1. Onchain Development

Scope

1. Directory Linked List:

- Maintain an operator registry with support for bond locking and governance mechanisms.

2. Midgard State Queue:

- A FIFO queue to store Layer 2 blocks
- This set of contracts on the Cardano L1 manages the true state of the L2. Each new Midgard block is appended to this list, resulting in a literal chain of Midgard blocks which lives on and is secured by Cardano.
- Metadata validation, such as block headers and state roots, to ensure data integrity.

3. Deposit/Withdraw Contracts:

- Enable secure uncensorable asset transfer between Layer 1 and Layer 2.

4. Fraud Proof Contracts:

- Modular contracts to ensure the security of Layer 2 state transitions. If a block contains any transaction which violates a ledger rule, a fraud proof can be constructed via these contracts, and the proof can then be used to invalidate the block and punish the associated operator by slashing their bond and removing them from the operator set.

Timeline

Q1 2025:

- Directory Linked List: Design and deploy functional operator registry contracts.
- Midgard State Queue: Develop and test FIFO mechanisms for Layer 2 blocks.
- Deposit/Withdraw Contracts: Build asset transfer contracts with proof verification.
- Fraud Proof Contracts (Phase 1): Implement fraud proofs for a core set of ledger rules.

Q2 2025:

- Fraud Proof Contracts (Phase 2): Complete the implementation of remaining ledger rules.

Estimated resources required

Smart contract developers: 5 FTE



MIDGARD

7.2. Offchain SDK Development

Scope

The offchain development focuses on creating the tools and logic required to construct and manage transactions that interact with the onchain contracts seamlessly.

Build SDKs to interact with :

- Directory Linked List:
 - Register operators.
 - Submit and update bond-locking transactions.
 - Interact with governance/protocol parameter mechanisms.
- Midgard State Queue:
 - Implement transaction building for submitting Layer 2 blocks to the onchain state queue.
 - Include metadata (e.g., block headers, state roots) to ensure proper validation.
 - Endpoints for queue monitoring (e.g., check incoming blocks, check canonical state).
- Deposit/Withdraw Contracts:
 - Develop transaction generators to:
 - Handle deposits by constructing proofs of state for Layer 1 to Layer 2 asset transitions.
 - Execute withdrawals using Layer 2 state proofs.
 -
- Fraud Proof Contracts:
 - Build a fraud-proof transaction generator for each rule.
 - Automate the submission process for detecting and reporting invalid blocks.
 - Create verification tools to ensure fraud proofs are formatted correctly before submission.
 - Develop a fraud proof step machine.

Estimated resources required

Offchain developer: 2 FTE

7.3. Midgard Layer 2 Node Development

Scope

The purpose of this component is to develop the Midgard Layer 2 node, responsible for interacting with the Midgard on-chain contracts and managing transactions, deposits, and withdrawals between Layer 1 and Layer 2. The main tasks include:

1. Midgard SDK Integration:

- Use the Midgard off-chain SDK to interface with Midgard on-chain contracts (e.g., deposit/withdrawal, state queue).
- Ensure seamless communication between the off-chain node and on-chain contracts, enabling transaction processing and block validation.

2. API Server Development:

- Set up an API server to receive and handle incoming transactions from users.
- Ensure secure, scalable API endpoints for transaction submission, balance queries, and withdrawal requests.

3. Transaction Validation:

- Validate transactions against Layer 2 ledger rules to ensure compliance before they are committed to the network.
- Implement robust logic to verify that transactions follow the correct sequence and comply with ledger rules, preventing violations such as double-spending.

Timeline

- Q1 2025:
 - Integrate the Midgard off-chain SDK with the Layer 2 node.
 - Develop the initial API server for transaction handling and communication with users.
 - Implement basic transaction validation and deposit/withdrawal functionalities.
- Q2 2025:
 - Refine and optimize transaction validation logic and ledger rule enforcement.
 - Implement robust deposit and withdrawal management between Layer 1 and Layer 2.
 - Conduct initial scalability and performance testing for transaction handling.

Estimated Resources Required

- Backend Developers: 3 FTE (For node, SDK integration, API, and transaction validation development).

7.4. Tx Builder Modification for Midgard Layer 2

Scope

The purpose of this component is to modify the current Tx Builder used in Cardano to accommodate the Midgard Layer 2 network. Midgard will be a subset of the Tx Builder, with a simplified transaction structure that omits certain fields not required by the Layer 2.

1. Transaction Structure Modification:

- Modify the Tx Builder to create transactions with a structure compatible with Midgard Layer 2.
- Ensure that unnecessary fields (e.g., Layer 1-specific fields) are excluded from the transaction.

2. Testing and Integration:

- Test the modified Tx Builder with Midgard Layer 2 contracts to ensure that transactions are properly formed, signed, and validated.
- Integrate the new Tx Builder with the Midgard Layer 2 node and verify that the transactions are processed according to the Layer 2 rules.

Timeline

- Q1 2025:
 - Review the existing Tx Builder code and identify the necessary modifications for Midgard Layer 2.
 - Implement changes to exclude unnecessary fields and add Midgard-specific fields.
 - Begin integrating the modified Tx Builder with the Midgard Layer 2 node.

Estimated Resources Required

Blockchain Developers: 2 FTE (For Tx Builder modification and integration with Midgard Node).

7.5 Testnet Deployment - DevOps/Distributed System Specialist

Scope

The goal is to set up and maintain a distributed Layer 2 blockchain environment for Midgard. This involves the following key responsibilities:

1. Testnet Setup:

- Provision hardware and services to create a distributed testnet.
- Deploy 1-2 Midgard nodes across two geographic locations to simulate network conditions.
- Ensure synchronization and fault tolerance between nodes.

2. Tooling and API Development:

- Develop APIs for node management, transaction submission, and block validation.
- Build operator interfaces to monitor and control the network.

3. CI/CD Implementation:

- Set up continuous integration and deployment pipelines for smart contract, tooling, and infrastructure updates.
- Automate testnet node provisioning and configuration management using Infrastructure as Code (IaC).

4. Testing & Benchmarking:

- Conduct periodic integration tests, load testing, and performance benchmarking.
- Automate regression and fault tolerance testing to simulate network failures, high traffic, and geographic latency.

Timeline

- Q1/Q2 2025:
 - Set up the testnet with 1-2 Midgard nodes across two geographic locations.
 - Develop initial APIs and operator interfaces.
 - Begin implementing CI/CD pipelines.

Resources Required

- DevOps/Distributed Engineers: 1 FTE
- Cloud Budget: ~\$3,000–\$5,000/month

7.6 Data Availability (DA) Layer Implementation

Scope

The purpose of this component is to develop the Data Availability (DA) layer for the Midgard Layer 2 network. For the MVP, the DA layer will store and manage temporary data using PostgreSQL.

1. DA Layer Architecture:

- Design and implement the architecture for the Data Availability layer, ensuring it can efficiently store and retrieve transaction data for Layer 2 operations.

2. Data Storage and Retrieval:

- Store incoming transaction data, state changes, and block metadata in PostgreSQL.

3. Transaction Indexing:

- Develop an indexing system for fast retrieval of transaction data, state roots, and other relevant information.
- Ensure that queries are optimized for performance to handle increasing volumes of data as the network scales.

4. Integration with Layer 2 Node:

- Ensure smooth interaction between the DA layer and the Midgard Layer 2 node.
- Allow the Layer 2 node to store and query relevant data from PostgreSQL for transaction validation and block creation.

5. Scalability Considerations:

- While using PostgreSQL for the MVP, we are planning to design the DA layer with security and scalability in mind.
- Plan for potential migration to a more distributed or blockchain-specific storage solution as the network grows.



Timeline

Q1 2025:

- Design the DA layer architecture and begin implementing PostgreSQL integration for temporary data storage.
- Develop basic data storage mechanisms for transaction metadata, state roots, and block headers.

Q2 2025:

- Optimize data retrieval, indexing, and query performance.
- Implement data integrity and consistency checks, ensuring that the data remains accurate and reliable.
- Begin integrating the DA layer with the Midgard Layer 2 node for data processing.

Estimated Resources Required

- Backend Developers: 2 FTE (For DA layer architecture, database integration, and transaction handling) + Cloud Budget: ~\$2,000–\$3,000/month for database infrastructure and scaling.



MIDGARD

8. Conclusion

The development of Midgard Layer 2 is progressing with a clear, structured plan across all key areas, from on-chain contract development and off-chain transaction handling, to the establishment of a robust Data Availability layer and the integration of a modified Tx Builder.

We are extremely optimistic about the successful delivery of this product. Our team consists of experts and talented developers with extensive experience in Cardano blockchain development. With our in-depth knowledge and years of hands-on experience working on critical Cardano protocols, we have thoroughly researched and validated the vision for Midgard Layer 2 to ensure it meets the high standards expected for such a system.

Moreover, we are following an iterative development process, working in sprints to identify potential issues early, gather valuable feedback, and continuously improve the implementation. This approach ensures that any challenges are addressed promptly, and we can deliver a robust, production-ready solution in alignment with the project's timeline.

We are confident that, with our expertise, commitment, and the iterative feedback loop, the Midgard Layer 2 will be a success and a valuable addition to the Cardano ecosystem.



MIDGARD